REMARKS

In a first Office Action dated September 9, 2003 (paper no. 3), the Examiner objected to claim 7 based on an imformality. The Examiner rejected claims 1-6 under 35 U.S.C. §101 as being directed to an algorithm and not embedded in a computer readable medium. The Examiner rejected claims 7-19 under 35 U.S.C. §103(a) as being unpatentable over Eroz et al. (U.S. patent no. 6,334,197, hereinafter referred to as "Eroz"). The rejections and objections are traversed and reconsideration is hereby respectfully requested.

The Examiner rejected claims 1-6 under 35 U.S.C. §101 as being directed to an algorithm and not embedded in a computer readable medium, noting that the steps therein are directed to mathematical algorithms rather than limited to practical applications. The applicants respectfully disagree. Claims 1-6 teach a method of interleaving data for Turbo Codes. The steps featured in these claims may be implemented in an application specific integrated circuit (ASIC) or in a digital signal processor (DSP). In case an ASIC approach is used, then hardware functions using combinatorial logic and finite state machines may implement the steps of Claims 1-6. In case a DSP is used, then assembly code is used to implement the steps in Claims 1-6 in the DSP. Therefore, the steps in Claims 1-6 are not directed to a mere mathematical algorithm but are rather a practical approach for interleaving bits in concatenated coding schemes, which approach may be implemented via hardware in an ASIC or via software in a DSP. Accordingly, the applicants respectfully request that the Examiner withdraw the rejection of claims 1-6 under 35 U.S.C. §101.

The Examiner rejected claims 7-19 under 35 U.S.C. §103(a) as being unpatentable over Eroz. Specifically, with respect to claims 7 and 14, the Examiner contended that Eroz teaches a two dimensional block interleaver (element 16 in FIG. 2) comprising a number of rows and a number of columns, wherein data is read into the interleaver row-by-row, then row and column permutations are performed to randomize data positions, and then the data is read out column-by-column (col. 9, lines 2-10). Specifically, the Examiner noted that Eroz teaches for an input position `1 = C * i + j,' a corresponding

output position is '$I(1) = R^*\Pi_i(j) + \rho(i)$,' where '$\Pi_i$' is a column permutation application to data in row I and 'ρ' is bit reversed indexing (col. 9, lines 11-17). The Examiner acknowledged that Eroz does not disclose a controller configured to bit reverse row and column indices. However, the Examiner contended that it would have been well known to one of ordinary skill in the art that controllers are required in order to perform read and write operations for forward or backward (bit reversing) operations.

The applicants respectfully disagree with the Examiner's interpretation of Eroz. In the sections of Eroz cited by the Examiner, Eroz teaches that, with respect to data read into an interleaver matrix, data is read out of the interleaver by first applying a "column permutation" to the columns of data in each row of the matrix. Eroz teaches column permutations that are applied on a row-by-row basis, that is, to the data in one row at a time, based on "a novel class of permutations derived from Galois Field arithmetic," with the result that a complicated set of logarithmic-based formulas are applied, bit-by-bit, to each bit position in a row in order to rearrange the order of the data in the row (col. 9, line 49, to col. 11, line 48). Bit reversed indexing is then applied to the reordered rows and the data is read out column-by-column.

This is completely different from the interleaving described in each of claims 1, 7, 13, and 14. Claims 1, 7, 13, and 14 teach bit-reversing the rows indices to reorder the rows, bit reversing the column indices to reorder the columns, then a systematic shifting of each row, that is, a shift of each row by none, one, or more steps. Data may then be read out after the shift of values in each row. This is much simpler than Eroz. Nowhere does Eroz propose bit reversing the column indices to reorder the columns. In fact, nowhere does the interleaver of Eroz operate on columns as a whole. The equations of Eroz reorders the data in a particular column by reodering the order of data in each row, on a row-by-row basis and bit-by-bit. By contrast, claims 1, 7, and 14 teach an interleaver that operates on entire rows (bit-reversing the rows indices to reorder the rows), on entire columns (bit reversing the column indices to reorder the columns), then on entire rows again (shifting each row none, one, or more steps).

Furthermore, claims 1, 7, and 14 reorder an interleaving matrix in three steps whereas Eroz reorders an interleaving matrix in two steps, with a very complex first step

(reordering the data in each row, on a row-by-row basis, followed by a second step of reordering the rows). As a result, despite teaching three reordering steps, claims 1, 7, 13, and 14 teach a much simpler, cleaner process of interleaving that does not involve complex formulas or Field theory and imposes a much lighter computational load on an associated processor. Therefore, it is no surprise that the relationship between the starting interleaver matrix (the matrix on page 6, line 12 of the pending application) (matrix 8 in Eroz) and the final interleaver matrix (the matrix on page 7, line 9 of the pending application) (matrix 17 in Eroz) is completely different between the pending application and Eroz and results in different final distances of adjacent input values. In fact, the entire emphasis of the pending application and the teachings of Eroz are different, as the interleaver of Eroz is designed to introduce a randomness into the separation of adjacent input bits while the interleaving taught by claims 1, 7, 13, and 14 is designed to deliberately, maximally separate adjacent input bits.

Therefore, Eroz does not teach the limitations of claims 1, 7, 13, and 14 of bit reversing row indexes for a first predetermined number of rows and permuting the corresponding data elements, bit reversing column indexes for a second predetermined number of columns and permuting the corresponding data elements, and shifting bit storage locations of one or more of the first predetermined number of rows a respective predetermined number of columns. Accordingly, the applicants respectfully request that claims 1, 7, 13, and 14 may now be passed to allowance.

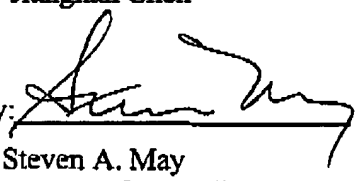
Since claims 2-6 depend upon allowable claim 1, claims 8-12 depend upon allowable claim 7, and claims 15-19 depend upon allowable claim 14, the applicants respectfully request that claims 2-6, 8-12, and 15-19 may now be passed to allowance.

The applicants have amended claim 7 as suggested by the Examiner in order to correct the informality.

As the applicants have overcome all substantive rejections and objections given by the Examiner and have complied with all requests properly presented by the Examiner, the applicants contend that this Amendment, with the above discussion, overcomes the Examiner's objections to and rejections of the pending claims. Therefore, the applicants

respectfully solicit allowance of the application. If the Examiner is of the opinion that any issues regarding the status of the claims remain after this response, the Examiner is invited to contact the undersigned representative to expedite resolution of the matter.

Respectfully submitted,
Jiangnan Chen

By: 

Steven A. May
Attorney for Applicants
Registration No. 44,912
Phone No.: 847/576-3635
Fax No.: 847/576-3750